

```

1
2
3 from a0_items import *
4 from plg_Regex import *
5 from selenium import webdriver
6 from fake_useragent import UserAgent
7 import time
8 from plg_Random import *
9
10
11 '''
12 1. Use chrome driver to find all elements of Home page - create a list for the elements
   ( LIST_element)
13
14 - get by id, by class, by name, create 3 separate lists and compare the lists and remove
   all duplicate values
15
16 - create one list from all 3 lists combined (id, class and name)
17
18 logic 1
19 1. Input search query into each element of the LIST_element by id, class, name...etc,
   then hit submit or return button
20 2. check if the returned page source is the correct search result page, containing the
   searched results - urls, content, keywords..etc
21 3. check if the url returned is the correct url containing correct search query
22
23 5. if none of the elements exists, or if none returned correct search results, go to
   next set of logic
24
25 logic 2
26 1. find a button to click on
27 2. after clicking on the button, see if a textbox appears, and if the textbox element
   exists in LIST_element
28 3. input search query into the textbox and submit
29
30 4. check if the returned page source is the correct search result page, containing the
   searched results - urls, content, keywords..etc
31 5. check if the url returned is the correct url containing correct search query
32
33 6. If the searchbox is still not found after clicking on the search button, add this
   page url to list --> LIST_no_searchbox
34
35
36 '''
37
38 #####
39 #####
40
41
42 def use_Chromedriver (path_Chromedriver, url, headless_enabled, js_enabled,
   user_agent_enabled, proxy_ip_port):
43
44     ### Use options to make chrome browser headless
45     #options = Options()
46     chrome_options = webdriver.ChromeOptions()
47
48     chrome_options.add_argument("--start-maximized") #need to maximize the browser window
   to see all the form controls
49     chrome_options.add_argument("--disable-extensions")
50     chrome_options.add_argument(f'--proxy-server={proxy_ip_port}')
51
52
53     if 'yes' in headless_enabled:
54
55         print ('Headless enabled')
56

```

```

57     chrome_options.add_argument("--headless")
58     #chrome_options.add_argument("--disable-javascript") //Not working
59
60     else:
61
62         print ('Headless NOT enabled')
63
64
65     if 'yes' in js_enabled:
66
67         print ('Javascript enabled')
68
69     else:
70         print ('Javascript NOT enabled')
71
72     prefs = {}
73     prefs["webkit.webprefs.javascript_enabled"] = False
74     prefs["profile.content_settings.exceptions.javascript.*.setting"] = 2
75     prefs["profile.default_content_setting_values.javascript"] = 2
76     prefs["profile.managed_default_content_settings.javascript"] = 2
77
78     chrome_options.add_experimental_option("prefs", prefs)
79     chrome_options.add_experimental_option( "prefs",{
80         'profile.managed_default_content_settings.javascript': 2})
81     #options.headless = True
82
83     if 'yes' in user_agent_enabled:
84
85         ###Use fake user agents
86         ua = UserAgent()
87         userAgent = ua.random
88         chrome_options.add_argument(f'user-agent={userAgent}')
89
90         print ('user agent enabled --> ' + str(userAgent))
91
92     else:
93
94         print ('user agent NOT enabled')
95
96     Chromedriver = webdriver.Chrome(path_Chromedriver, options = chrome_options)
97     Chromedriver.get(url)
98     time.sleep(10)
99
100    return Chromedriver
101
102
103
104    #####
105    #####
106
107    def get_proxy_list ():
108
109        LIST_proxy_ip_port = []
110
111        #get GOOD proxy from DB_proxies
112
113        return LIST_proxy_ip_port
114
115
116
117
118    def get_page_source (path_Chromedriver, url):
119
120        LIST_proxy_ip_port = get_proxy_list()
121        random_proxy = random_select_from_list (LIST_item=LIST_proxy_ip_port)

```

```

122
123     d_Chrome = use_Chromedriver (path_Chromedriver, url, headless_enabled='no', js_enabled
124     ='yes', user_agent_enabled = 'yes', proxy_ip_port=random_proxy)
125
126     page_source = d_Chrome.page_source
127
128     #print (page_source.encode('utf-8'), flush=True)
129     return page_source
130
131 #get_page_source
132 (path_Chromedriver="../../data/raw/Chromedrivers/chromedriver-win64/chromedriver.exe",
133 url='https://duckduckgo.com/?t=h_&q=python&ia=web')
134
135 def get_ALL_links (path_Chromedriver, url):
136
137
138     html_page = get_page_source (path_Chromedriver=path_Chromedriver, url=url)
139     LIST_link = []
140
141     soup = BeautifulSoup(html_page)
142
143     for link in soup.findAll(attrs={'href': re.compile("http")}):
144
145         http_link = link.get('href')
146         print (http_link, flush=True)
147
148         LIST_link.append(http_link)
149
150     return LIST_link
151
152
153 #get_ALL_links
154 (path_Chromedriver="../../data/raw/Chromedrivers/chromedriver-win64/chromedriver.exe",
155 url='https://duckduckgo.com/?t=h_&q=python&ia=web', proxy_ip_port='')
156
157
158 # Xpath_value = "//*[@id]"      Xpath_value = "//*[@id='query']"
159 # Xpath_value = "//*[@class]"
160 def get_ALL_attributes_by_xpath (path_Chromedriver, url, Xpath_value):
161
162     LIST_html_attribute = []
163
164     LIST_proxy_ip_port = get_proxy_list()
165     random_proxy = random_select_from_list (LIST_item=LIST_proxy_ip_port)
166
167     d_Chrome = use_Chromedriver (path_Chromedriver, url, headless_enabled='no', js_enabled
168     ='no', user_agent_enabled = 'yes', proxy_ip_port=random_proxy)
169
170     try:
171
172         #https://stackoverflow.com/questions/20244691/python-selenium-how-do-i-find-all-ele
173         #ment-ids-on-a-page
174         LIST_element = d_Chrome.find_elements(By.XPATH, Xpath_value)
175         # to get names use "//*[@name]"
176
177         for element in LIST_element:
178
179             LIST_html_attribute.append('Tag: ' + str(element.tag_name).encode('ascii',
180             'ignore').decode('ascii')
181             + ' ID: ' + str(element.get_attribute('id')).encode('ascii', 'ignore').decode(

```

```

180         'ascii')
181         + ' Name: ' + str(element.get_attribute('name')).encode('ascii', 'ignore').
182         decode('ascii')
183         + ' Type: ' + str(element.get_attribute('type')).encode('ascii', 'ignore').
184         decode('ascii')
185         + ' Title: ' + str(element.get_attribute('title')).encode('ascii', 'ignore').
186         decode('ascii')
187         + ' Class: ' + str(element.get_attribute('class')).encode('ascii', 'ignore').
188         decode('ascii')
189         + ' Placeholder: ' + str(element.get_attribute('placeholder')).encode('ascii',
190         'ignore').decode('ascii'))
191
192     print ('LIST html attribute => ' + str(LIST_html_attribute))
193     return LIST_html_attribute
194
195 except:
196
197     print ('Elements NOT found')
198
199 #LIST_output = None
200
201 #LIST_output = get_ALL_attributes_by_xpath
202 (path_Chromedriver=" ../data/raw/Chromedrivers/chromedriver-win64/chromedriver.exe",
203 url="https://timesofindia.indiatimes.com/", Xpath_value= "//*[@id]")
204 LIST_output = get_ALL_attributes_by_xpath (path_Chromedriver=
205 " ../data/raw/Chromedrivers/chromedriver-win64/chromedriver.exe", url=
206 "https://timesofindia.indiatimes.com/", Xpath_value= "//*[@class]")
207 #LIST_output = get_ALL_attributes_by_xpath
208 (path_Chromedriver=" ../data/raw/Chromedrivers/chromedriver-win64/chromedriver.exe",
209 url="https://timesofindia.indiatimes.com/", Xpath_value= "//*[@name]")
210 #LIST_output = get_ALL_attributes_by_xpath
211 (path_Chromedriver=" ../data/raw/Chromedrivers/chromedriver-win64/chromedriver.exe",
212 url="https://timesofindia.indiatimes.com/", Xpath_value= "//*[@type]")
213 #LIST_output = get_ALL_attributes_by_xpath
214 (path_Chromedriver=" ../data/raw/Chromedrivers/chromedriver-win64/chromedriver.exe",
215 url="https://timesofindia.indiatimes.com/", Xpath_value= "//*[@aria-label]")
216
217 '''
218 #
219 https://rollbar.com/blog/python-typeerror-nonetype-object-is-not-iterable/#:~:text=The%20
220 Python%20TypeError%3A%20NoneType%20Object%20Is%20Not%20Iterable%20error%20can,over%2C%20w
221 hich%20avoids%20the%20error.
222 if LIST_output is not None:
223
224     for x in LIST_output:
225
226         if 'OGLTB' in x:
227
228             print (x, flush=True)
229
230         else:
231
232             pass
233
234 else:
235
236     print ('LIST has NO elements')
237 '''
238
239 #####
240 #####
241
242 def input_search_bar(path_Chromedriver, url, Xpath_value, search_query, remove_after_text
243 ):

```

```

226
227
228 LIST_proxy_ip_port = get_proxy_list()
229 random_proxy = random_select_from_list (LIST_item=LIST_proy_ip_port)
230
231 d_Chrome = use_Chromedriver (path_Chromedriver, url, headless_enabled='no', js_enabled
232 ='yes', user_agent_enabled='yes', proxy_ip_port=random_proxy)
233 print(d_Chrome.title)
234
235 try:
236     search_bar = d_Chrome.find_element(By.XPATH, Xpath_value)
237     search_bar.clear()
238
239     search_bar.send_keys(search_query)
240     search_bar.send_keys(Keys.RETURN)
241
242     time.sleep(5)
243
244     full_url = d_Chrome.current_url
245     Domain_section_query_url = remove_all_after_text (after_text=remove_after_text,
246 text_main= full_url)
247
248     print(full_url)
249     print (Domain_search_query_url)
250
251     return full_url, Domain_section_query_url
252
253 except:
254
255     print('Not inputable')
256
257 #input_search_bar (path_Chromedriver =
258 ".../data/raw/Chromedrivers/chromedriver-win64/chromedriver.exe",
259 url='https://google.com', Xpath_value = '//*[@name="q"]', search_query='python coding',
260 remove_after_text='python', proxy_ip_port='')
261
262 #####
263 #####
264
265 def click_on_button (path_Chromedriver, url, Xpath_value):
266
267     LIST_proxy_ip_port = get_proxy_list()
268     random_proxy = random_select_from_list (LIST_item=LIST_proy_ip_port)
269
270     d_Chrome = use_Chromedriver (path_Chromedriver, url, headless_enabled='no', js_enabled
271 ='yes',user_agent_enabled='yes', proxy_ip_port =random_proxy)
272
273     try:
274         button = d_Chrome.find_element(By.XPATH, Xpath_value)
275
276         button.click()
277         time.sleep(5)
278
279         print('Clickable')
280
281         return button
282
283     except:
284
285         print('Not clickable')
286
287 #click_on_button (path_Chromedriver =
288 ".../data/raw/Chromedrivers/chromedriver-win64/chromedriver.exe",
289 url='https://timesofindia.indiatimes.com/', Xpath_value = "//*[@class='OG1TB']",

```

```

284 proxy_ip_port='')
285
286
287 #####
288
289 ## test proxy
290 #####
291
292 def test_proxy (LIST_proxy, url, proxy_ip_port):
293
294
295     random_proxy = random.choice(PROXIES)
296
297     seleniumwire_options = {
298         'proxy': {
299             'http': f'{random_proxy}',
300             'https':f' {random_proxy}',
301             'verify_ssl': False,
302         },
303     }
304
305
306     Chromedriver = webdriver.Chrome (
307         service=ChromeService(ChromeDriverManager().install()),
308         seleniumwire_options=seleniumwire_options
309     )
310
311     Chromedriver.visit()
312
313
314 proxies = ['http://103.21.244.160:80', 'http://132.226.118.106:8080' ]
315 #test_proxy (LIST_proxy, url='https://timesofindia.indiatimes.com/')
316
317
318
319 def test_proxy_1 (path_Chromedriver, proxy):
320
321     chrome_options = webdriver.ChromeOptions()
322     #chrome_options.add_argument('--proxy-server=%s' % PROXY)
323     chrome_options.add_argument(f'--proxy-server={PROXY} ')
324
325     chrome_options.add_argument("ignore-certificate-errors")
326
327     chrome = webdriver.Chrome(path_Chromedriver, options=chrome_options)
328     #chrome.get("https://www.ipchicken.com/")
329     chrome.get("https://api.ipify.org/?format=json")
330
331     time.sleep(20)
332
333
334 #test_proxy_1
335 (path_Chromedriver="../../../data/raw/Chromedrivers/chromedriver-win64/chromedriver.exe",
336 proxy = "11.456.448.110:8080" # HOST:PORT)
337
338 #####
339 ## Test list of elements
340
341
342 #https://stackoverflow.com/questions/52706901/how-to-assert-the-text-box-is-enabled-or-di
343 #sabled
344 #https://pythonexamples.org/python-selenium-check-if-input-text-field-exists/
345 #https://www.browserstack.com/guide/python-selenium-to-run-web-automation-test

```

```

346
347 def input_search_bar_LIST (path_Chromedriver, url, Xpath_value, search_query):
348
349     d_Chrome = get_HTML_attributes_by_xpath (path_Chromedriver, url, Xpath_value)[1]
350     LIST_attribute = get_HTML_attributes_by_xpath (path_Chromedriver, url, Xpath_value)[0]
351
352     for attribute in LIST_attribute:
353
354         #get the ID only
355         ID = remove_all_before_and_after_texts (text_1 = 'ID:', text_2='Name:', text =
            attribute)
356
357         # Check if value in input text is empty
358         try:
359
360             id_element = d_Chrome.find_elements(By.XPATH, Xpath_v = "//*[@id=" + ID + "]" )
361             #print("Input text field exists.")
362
363             id_element.clear()
364
365             id_element.send_keys(search_query)
366
367             id_element.submit()
368             time.sleep(5)
369
370             full_url = Chromedriver.current_url
371             print(full_url)
372
373             Domain_search_query_url = remove_all_after_text (after_text=search_query,
                text_main= full_url)
374
375             print ('dsqu = ' + Domain_search_query_url)
376
377
378         except (NoSuchElementException, WebDriverException):
379
380             print("Input textbox NOT exists.")
381
382         # Close the driver
383         #Chromedriver.quit()
384
385
386 #input_search_textbox (path_Chromedriver =
    "../../data/raw/Chromedrivers/chromedriver-win64/chromedriver.exe",
    url="https://timesofindia.indiatimes.com/", Xpath_value= "//*[@id]", search_query =
    'orange')
387 #input_search_textbox (path_Chromedriver =
    "../../data/raw/Chromedrivers/chromedriver-win64/chromedriver.exe",
    url="https://google.com/", Xpath_value="//*[@name='q']")
388
389
390
391
392 def click_on_button_LIST ():
393
394     d_Chrome = get_HTML_attributes_by_xpath (path_Chromedriver, url, Xpath_value)[1]
395     LIST_attribute = get_HTML_attributes_by_xpath (path_Chromedriver, url, Xpath_value)[0]
396
397     pass
398
399
400
401
402 def get_page_source_URLS ():
403
404     pass

```